## REMARKS

Reconsideration of the above-identified patent application in view of the amendments above and the remarks following is respectfully requested.

Claims 1-16 are in this case. Claims 6-13 have been rejected under § 112, second paragraph. Claims 1-4 and 6-16 have been rejected under § 102(e). Claim 5 has been rejected under § 103(a). Claims 14 and 15 have been objected to. Claims 2, 7-9, 11, 12 and 15 have been canceled. Independent claims 1, 13 and 14 and dependent claims 6 and 10 have been amended. New claims 17-28 have been added.

The claims before the Examiner are directed toward a non-volatile memory device for storing both code and data. While the device is programming or erasing its nonvolatile memory, if a host system sends the device a read request, hardware such as one or more logic circuits in the device signals the host to delay executing the request while the device suspends the programming/erasing. Then the hardware suspends the programming/erasing and signals the host to execute the read request. When all read requests have concluded, the hardware resumes the programming/erasing.

### § 112, Second Paragraph Rejections

The Examiner has rejected claims 6-13 under § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Specifically, the Examiner has pointed out that claims 6, 7, 10, 12 and 13 recite the limitation "the/said memory chip" which has insufficient antecedent basis; that claims 8 and 9 recite the limitation "said logic circuits" (plural) which has

insufficient antecedent basis; and that claim 11 recites the limitation "the OS/application/file management software", which has insufficient antecedent basis.

Claims 7-9, 11 and 12 have been canceled, thereby rendering moot the Examiner's rejection of these claims.

Claim 6 has been amended to recite "the memory device" instead of "the memory chip". "The memory device" has antecedent basis in the preamble of claim 1, from which claim 6 depends. Support for this amendment is found in the specification in Figure 4, that shows two logic circuits for automatic suspending and resuming of operations, automatic suspend logic **26** and automatic resume logic **27**, inside the rectangle that encloses the other components (Bus I/F logic **23**, Flash array **24** and Flash circuitry **25**) of the memory device.

Claim 10 has been amended to recite "said memory device" instead of "said memory chip. "The memory device" has antecedent basis in the preamble of claim 1, from which claim 10 depends. Support for this amendment is found in the specification on page 8 lines 16-19:

> The automatic suspend logic **26** (Fig. 4) is operated when an erase or program operation begins **15** (Fig. 3). When detecting one of these operations (erase or program) the automatic suspend logic **26** is triggered. From this moment onwards, the logic waits for a read operation **11** from the device (read operations that requires the device to output real data as opposed to status bits or similar). (emphasis added)

**§ 102(e) Rejections – See et al. '070**

The Examiner has rejected claims 1-4 and 6-16 under § 102(e) as being anticipated by See et al., US Patent No. 6,189,070 (henceforth, "See et al. '070"). The Examiner's rejection is respectfully traversed.

Claims 2, 7-9, 11, 12 and 15 have been canceled, thereby rendering moot the Examiner's rejection of these claims.

Like the present invention, See et al. '070 teach a memory device **410** whose programming/erasing operations are suspended when host system **400** that accesses device **410** needs to read data from device **410**. Unlike the present invention, See et al. '070 rely on software (for example the operating system of the host system), rather than on hardware of device **410** itself, to initiate suspension and resumption of the programming/erasing operations as needed.

That See et al. '070 rely on software is clear from the introductory paragraph of the operational details of their preferred embodiment, in column 5 lines 1-10:

> The present invention, instead, uses a method of preventing the processor from automatically vectoring to the flash memory in response to a system interrupt **200**. This solution disables interrupts when a non-read operation (e.g., a program operation or an erase operation) is performed on the flash device **410**. Interrupts are then checked, e.g., by polling. If an interrupt is found, then the non-read operation is suspended in the flash device. Interrupts are enabled so that the processor can vector to the flash memory in order to executed code located in the flash memory. (emphasis added)

Disabling and enabling interrupts are operations that would be done by the operating system of system **400**.

That See et al. '070 rely on software also is clear from some of the subsequent operational details. For example, column 5 lines 11-21:

> FIGS. 4a and 4b are a flowchart of the present invention showing exemplary steps taken by a system **400** having the configuration shown in FIGS. 3a, 3b, and 3c in performing non-read operations stored in the volatile memory **405**. The flow chart starts at block **500**. The operation continues at block **502**, at which task scheduling is disabled. The operation continues at block **504**, at which interrupts are disabled. In one embodiment, the disabling of task scheduling and interrupts is performed by setting a flag or register within the processor **402**. From block **504** operation continues at block **510**.

Disabling task scheduling and interrupts are operations that would be done by the operating system of system **400**. In the specific embodiment described in this paragraph, these operations are done by processor **402**, not by memory device **410**.

Specifically with regard to the suspension and resumption of programming and erasing, See et al. '070 provide for eight commands to memory device **410** from system **400** (column 7 lines 50-52):

> (1) erase, (2) erase suspend, (3) erase resume, (4) program, (5) program suspend, (6) program resume, (7) read, and (8) read status.

as opposed to the prior art command set (column 6 lines 60-62) that lacks the program suspend and program resume commands. The need for the two additional commands is evident from column 6 lines 8-10:

> At block **542** the program or erase operation being executed in the non-volatile writeable memory is suspended.

and from the flow of control from the "Yes" branch of block **556** of Figure 4B ultimately back to block **520** of Figure 4B. If an erase or program operation was suspended in block **542**, then the operation needs to be resumed in block **520**. In block **542**, system **400** sends an "erase suspend" or a "program suspend" command to memory device **410**. In block **520**, system **400** sends an "erase resume" or a "program resume" command to memory device **410**.

By contrast, in the present invention, the hardware of the memory device does all the work of suspending and resuming erase and program operations, including initiating the suspending and resuming of erase and program operations, in response only to read commands received from the host system, with no participation by the host system except for refraining from trying to read data from the memory device while the memory device is entering suspended mode. Such autonomous operation of a memory device, without the use of explicit erase/program suspend/resume commands from a host system, is neither taught by See et al. '070 nor obvious from See et al. '070.

While continuing to traverse the Examiner's rejections, Applicant has, in order to expedite the prosecution, chosen to amend independent claims 1, 13 and 14 in order to clarify and emphasize the crucial distinctions between the present invention and the teachings of See et al. '070. Specifically:

Claim 1 has been amended to state that the logic circuit is separate from the CPU/Bus/Controller (now called the "host": see "Other Amendments to the Claims" below) and that the automatic suspending/resuming is in response to a read request from the CPU/Bus/Controller. Support for the logic circuit being separate from the CPU/Bus/Controller is found in the specification in Figure 4, that shows automatic suspend logic **26** and automatic resume logic **27** separate from CPU Bus **20**. Support for the suspending/resuming being in response to a read request is found in the specification on page 9 lines 1-2:

> Upon detection of the read operation **11** the automatic suspend logic **26** executes a process that enters the device into the suspend state **12**.

Note that read operation **11** is called a "read request" on page 6 line 19 of the specification.

Claim 13 has been amended to state that the monitoring of the status of current operations, the signaling if the device is available for code execution, and the commanding to suspend/resume operations are effected by the at least one logic circuit; and that the commanding to suspend/resume operations is in response to a read request. Support for the monitoring of the status of current operations being effected by the at least one logic circuit is found in the specification on page 8 lines 16-20:

> The automatic suspend logic **26** (Fig 4) is operated when an erase or program operation begins **15** (Fig 3). When detecting one of these operations (erase or program) the automatic suspend logic **26** is triggered. From this moment onwards, the logic waits for a read

operation **11** from the device (read operations that require the device to output real data as opposed to status bits or similar).

Support for the signaling if the device is available for code execution being effected by the at least one logic circuit is found in the specification on page 9 lines 10-14:

> The logic is also responsible of verifying that the device has actually entered the automatic suspend state **12**. After the verification phase – the Busy signal **22** will be turned off (to indicate that the device has entered the automatic suspend state **12**.

Support for the commanding to suspend/resume operations being effected by the at least one logic circuit in response to a read request is found in the specification on page 9 lines 1-2:

> Upon detection of the read operation **11** the automatic suspend logic **26** executes a process that enters the device into the suspend state **12**.

(note that read operation **11** is called a "read request" on page 6 line 19 of the specification), page 9 lines 16-17:

> The automatic resume logic **27** starts to operate when the device enters the automatic suspend state **12**.

and page 9 line 21 through page 10 line 1:

> One suggested implementation is to wait for a break in the read operations of the device. When the break is long enough...the logic executes a process which causes the device to resume the program/erase operation **15**...

In addition, the step of monitoring CPU/Bus activity has been deleted from claim 13 because it is not necessary to recite this step to distinguish the present invention from the prior art; and "to the CPU/Bus" has been deleted from the signaling step because "the CPU/Bus" lacks antecedent basis.

Claim 14 has been amended to state that the sensing of read requests (now amended to sensing of a read request: see "Other Amendments to the Claims" below), the entering into suspended mode, the signaling to CPU/Bus, the turning off of the signal and the entering of the device into resume operations (now amended to exiting

from suspended mode: see "Other Amendments to the Claims" below) are effected by the at least one logic circuit; and that the entering into suspended mode is in response to the sensing of a read request. Support for the sensing of a read request being effected by the at least one logic circuit is found in the specification on page 8 lines 16-20:

> The automatic suspend logic **26** (Fig 4) is operated when an erase or program operation begins **15** (Fig 3). When detecting one of these operations (erase or program) the automatic suspend logic **26** is triggered. From this moment onwards, the logic waits for a read operation **11** from the device (read operations that require the device to output real data as opposed to status bits or similar).

Support for the entering into suspended mode being effected by the at least one logic circuit in response to the sensing of a read request is found in the specification on page 9 lines 1-2:

> Upon detection of the read operation **11** the automatic suspend logic **26** executes a process that enters the device into the suspend state **12**.

Note that read operation **11** is called a "read request" on page 6 line 19 of the specification. Support in the specification for the signaling to CPU/Bus being effected by the at least one logic circuit is found in the specification on page 9 lines 7-10:

> In addition, the logic will indicate that the device is on its way to the automatic suspend state **12** using an external signal (Busy signal) **22**. This signal can be used by the platform to hold/retry the read operation **11** attempt or any other mechanism in the CPU/Bus that can delay execution of read/fetch cycles.

Support in the specification for the turning off of the signal being effected by the at least one logic circuit is found in the specification on page 9 lines 11-14:

> After the verification phase – the Busy signal **22** will be turned off (to indicate that the device has entered the automatic suspend state **12**. From this moment onward the device is ready to perform read requests ad required.

Support for the exiting from suspended mode being effected by the at least one logic circuit is found in the specification on page 9 line 16 through page 10 line 1:

> The automatic resume logic **27** starts to operate when the device enters the automatic suspend state **12**. The target of this logic is to resume the program/erase operation **15** that was interrupted by the automatic suspend logic **26**. This logic should monitor the read operations done from the device, for example, by using the same techniques as the automatic suspend logic **26**. The logic is responsible to resume the suspended operation. One suggested implementation is to wait for a break in the read operations of the device. When the break is long enough...the logic executes a process which causes the device to resume the program/erase operation **15**...

Amended independent claims 1, 13 and 14 now feature language which makes it absolutely clear that the memory device of the present invention functions autonomously, under the control of its own logic circuit(s), to suspend erase/program operations in response to read requests and to subsequently resume those operations. Applicant believes that the amendment of the claims completely overcomes the Examiner's rejections on § 102(b) grounds.

With independent claims 1 and 14 allowable in their present form, it follows that claims 3, 4, 6 and 10, that depend therefrom, also are allowable.

Claim 16 is allowable over See et al. '070 as filed, by virtue of reciting hardware initiated suspending and resuming. In See et al. '070, the initiative for suspending and resuming comes from system **400**, not from memory device **410**.

**§ 103(a) Rejections – See et al. '070 and Keeley et al. '790**

The Examiner has rejected claims 5 under § 103(a) as being unpatentable over See et al. '070 and Keeley et al., US Patent No. 4,491,790. The Examiner's rejection is respectfully traversed.

It is demonstrated above that independent claim 1 is allowable in its present form. It follows that claim 5, that depends therefrom, also is allowable.

## Other Amendments to the Claims

Claims 1 and 14 have been amended to recite the present invention more precisely.

In claim 1, the "CPU/Bus/controller" now is called the "host". Support for this amendment is found in the specification on page 7 lines 10-11, in which CPU/Bus/Controller **20** is called "host CPU/Bus/Controller **20**".

Claim 1 also now recites the function of the host as being for "accessing" the memory device, rather than for "controlling" the memory device. This is a more accurate description of what is done by the entity that exchanges signals with the memory device of Figure 4 via CPU bus **20**.

In step ii of claim 14, "a read request" (singular) has been changed to "read requests" (plural). Support for this amendment is found in the specification on page 8 lines 18-19:

> From this moment onwards, the logic waits for a read operation **11** from the device... (emphasis added)

Note that read operation **11** is called a "read request" on page 6 line 19 of the specification.

In step iv of claim 14, signaling to "wait before executing further read/fetch commands" has been changed to signaling to "delay executing said read request". Support for this amendment is found in the specification on page 9 lines 7-10:

> In addition, the logic will indicate that the device is on its way to the automatic suspend state **12** using an external signal (Busy signal) **22**. This signal can be used by the platform to hold/retry the read operation **11** attempt or any other mechanism in the CPU/Bus that can delay execution of read/fetch cycles.

Note that read operation **11** is called a "read request" on page 6 line 19 of the specification.

In step v of claim 14, turning off the signal to allow the "continuation" of "read/fetch commands" has been changed to turning off the signal to allow the "execution" of "said read request". Support for this amendment is found in the specification on page 9 lines 11-14:

> After the verification phase – the Busy signal **22** will be turned off (to indicate that the device has entered the automatic suspend state **12**. From this moment onwards the device is ready to perform read requests as required.

In step vi of claim 14, "entering...into resume operations" has been changed to "exiting...from said suspended mode". Support for this amendment is found in the specification in Figure 3, in which arrow **13** represents exiting from suspend state **12** to resume state **14**.

## New Claims

New claims 17-28 have been added.

New claim 17 recites a memory device that comprises a non-volatile memory; circuitry for reading, programming and erasing the non-volatile memory; and a hardware mechanism for suspending an activity of the circuitry in response to at least one read request.

Support for the presence of the non-volatile memory in the memory device is found in the specification in Flash array **24** of Figure 4.

Support for the presence in the memory device of the circuitry for reading, programming and erasing the non-volatile memory is found in the specification in Flash circuitry **25** of Figure 4.

Support for the presence in the memory device of the hardware mechanism for suspending the activity of the circuitry in response to at least one read request is found

in the specification in Automatic suspend logic **26** of Figure 4, as described on page 9 lines 1-2:

> Upon detection of the read operation **11** the automatic suspend logic **26** executes a process that enters the device into the suspend state **12**.

Note that read operation **11** is called a "read request" on page 6 line 19 of the specification.

New claim 18 recites the limitation that the hardware mechanism resumes the activity after the circuitry has finished processing the at least one read request. Support for this limitation is found in the specification in Automatic resume logic **27** of Figure 4, as described on page 9 line 16 through page 10 line 3:

> The automatic resume logic **27** starts to operate when the device enters the automatic suspend state **12**. The target of this logic is to resume the program/erase operation **15** that was interrupted by the automatic suspend logic **26**. This logic should monitor the read operations done from the device, for example, by using the same techniques as the automatic suspend logic **26**. The logic is responsible to resume the suspended operation. One suggested implementation is to wait for a break in the read operations of the device. When the break is long enough...the logic executes a process which causes the device to resume the program/erase operation **15** (e.g. executing the resume command which is available in certain devices). The logic contains some mechanism to determine if the break is a real break or just a temporary break (e.g. a timer that counts the no-read-operation time).

New claim 19 limits the suspended/resumed activity to erasing the non-volatile memory. New claim 20 limits the suspended/resumed activity to programming the non-volatile memory. Support for these limitations is found in the specification on page 6 lines 14-16:

> The hardware mechanism of the present invention, which is one logic circuit (or a few circuits), is designed so as to enable automatic suspend and automatic resume of program and/or erase operations...

and on page 8 lines 17-18:

> When detecting one of these operations (erase or program) the automatic suspend logic **26** is triggered.

New claim 21 recites the limitation that the hardware mechanism includes at least one logic circuit. Support for this limitation is found in the specification on page 6 lines 14-15:

> The hardware mechanism of the present invention, which is one logic circuit (or a few circuits)...

New claim 22 recites the limitation that the suspending of the activity includes indicating to the source (host) of the at least one read request that the suspending has commenced and subsequently indicating to the host that the memory device is available for reading. Support for this limitation is found in the specification on page 9 lines 7-14:

> In addition, the logic will indicate that the device is on its way to the automatic suspend state **12** using an external signal (Busy signal) **22**. This signal can be used by the platform to hold/retry the read operation **11** attempt or any other mechanism in the CPU/Bus that can delay execution of read/fetch cycles. The logic is also responsible of verifying that the device has actually entered the automatic suspend state **12**. After the verification phase – the Busy signal **22** will be turned off (to indicate that the device has entered the automatic suspend state **12**. From this moment onward the device is ready to perform read requests ad required.

Note that read operation **11** is called a "read request" on page 6 line 19 of the specification.

New claim 23 recites the limitation that the hardware mechanism also monitors the processing of the at least one read request to determine when the circuitry has finished processing the at least one read request. Support for this limitation is found in the specification on page 9 line 18 through page 10 line 3:

> This logic should monitor the read operations done from the device, for example, by using the same techniques as the automatic suspend logic **26**. The logic is responsible to resume the suspended operation. One suggested implementation is to wait for a break in the read operations of the device. When the break is long enough...the logic executes a process which causes the device to resume the program/erase operation **15** (e.g. executing the resume command

which is available in certain devices). The logic contains some mechanism to determine if the break is a real break or just a temporary break (e.g. a timer that counts the no-read-operation time).

New claim 24 recites a method for managing a memory device that includes a non-volatile memory and that is accessed by a host. First, the memory device commences either erasing or programming the volatile memory. During the erase or program operation, the host requests a read operation. In response to the request, the memory device suspends the operation.

Support for the memory device including a non-volatile memory is found in the specification in Flash array **24** of Figure 4. Support for the memory device being accessed by a host is found in the specification via CPU bus **20** of Figure 4: the entity that sends commands to the memory device via CPU bus **20** is a host of the memory device. Indeed, this entity is called "host CPU/Bus/Controller" **20** on page 7 lines 10-11.

Support for commencing erasing or programming of the volatile memory is found in the specification on page 6 line 18:

The memory device is executing the erase/program operation **15**...

on page 7 line 5:

1.1 The device is busy with erase/program operation **15**.

and on page 8 lines 16-17:

The automatic suspend logic **26** (Fig 4) is operated when an erase or program operation begins **15** (Fig 3).

Support for the hose requesting a read operation while the memory device is erasing or programming is found in the specification on page 6 lines 18-19:

The memory device is executing the erase/program operation **15** and at the same time a read request **11** is registered.

Support for the memory device suspending the operation in response to the request is found in the specification on page 9 lines 1-2:

> Upon detection of the read operation **11** the automatic suspend logic **26** executes a process that enters the device into the suspend state **12**.

Note that read operation **11** is called a "read request" on page 6 line 19 of the specification.

New claim 25 recites the additional step of the memory device signaling to the host to delay the request, in response to the request. Support for this additional step is found in the specification on page 9 lines 7-10:

> In addition, the logic will indicate that the device is on its way to the automatic suspend state **12** using an external signal (Busy signal) **22**. This signal can be used by the platform to hold/retry the read operation **11** attempt or any other mechanism in the CPU/Bus that can delay execution of read/fetch cycles.

New claim 26 recites the additional step of the host delaying execution of the request in response to the signal from the memory device to delay the request. Support for this additional step is found in the specification on page 7 lines 10-13:

> The host CPU/Bus/Controller **20** or host Bus **21** uses this signal to hold/retry operations using its standard hold/retry mechanisms, or any other means provided by the CPU/Bus/Controller to prevent a crash due to a failed read attempt.

and on page 9 lines 8-10:

> This signal can be used by the platform to hold/retry the read operation **11** attempt or any other mechanism in the CPU/Bus that can delay execution of read/fetch cycles.

Note that read operation **11** is called a "read request" on page 6 line 19 of the specification.

New claim 27 recites the additional step of the memory device signaling the host to resume execution of the request. Support for the memory device signaling the

host to resume execution of the request is found in the specification on page 9 lines 11-14:

> The logic is also responsible of verifying that the device has actually entered the automatic suspend state **12**. After the verification phase – the Busy signal **22** will be turned off (to indicate that the device has entered the automatic suspend state **12**. From this moment onwards the device is ready to perform read requests as required.

New claim 28 recites the additional step of the memory device monitoring a conclusion of read requests from the host, subsequent to suspending the operation; and also the additional step of the memory device resuming the operation upon detection of the conclusion of read requests. Support for the memory device monitoring a conclusion of read requests from the host, subsequent to suspending the operation, is found in the specification on page 6 line 22 through page 7 line 1:

> The read operation will subsequently be monitored in order to detect a timeout **13** (a predetermined period of time in which no read operation is done.

and on page 9 lines 18-20:

> This logic should monitor the read operations done from the device, for example, by using the same techniques as the automatic suspend logic **26**.

Support for the memory device resuming the operation upon detection of the conclusion of read requests is found in the specification on page 7 lines 1-2:

> Upon detection it automatically gives a command to resume operations **14**, allowing the programming/erase operation **15** to continue.

and on page 9 line 20 through page 10 line 1:

> The logic is responsible to resume the suspended operation. One suggested implementation is to wait for a break in the read operations of the device. When the break is long enough...the logic executes a process which causes the device to resume the program/erase operation **15**...

## Objections to the Claims

The Examiner has objected to claim 14 because the word "automatically" is enclosed in parentheses. The word "automatically" and the enclosing parentheses have been deleted.

The Examiner has objected to claim 14 because the claim recites "the a non-volatile memory chip". This inadvertent typographical error has been corrected by deleting the extra "a".

The Examiner has objected to claim 15 for depending from itself. Claim 15 now has been canceled, thereby rendering moot the Examiner's rejection.

## Objections to the Drawings

The Examiner has objected to Figure 1 for lacking a legend such as "PRIOR ART" to indicate that only prior art is illustrated. Attached please find a replacement drawing in which both Figure 1 and Figure 2 are labeled as PRIOR ART in red.

The Examiner has objected to the drawings for not showing the external logic circuits recited in claims 7 and 9. Claims 7 and 9 have been canceled, thereby rendering moot the Examiner's objection.

In view of the above amendments and remarks it is respectfully submitted that independent claims 1, 13, 14, 16, 17 and 24, and hence dependent claims 3-6, 10, 18-23 and 25-28 are in condition for allowance. Prompt notice of allowance is respectfully and earnestly solicited.

Respectfully submitted,

Mark M. Friedman
Attorney for Applicant
Registration No. 33,883

Date: December 22, 2003

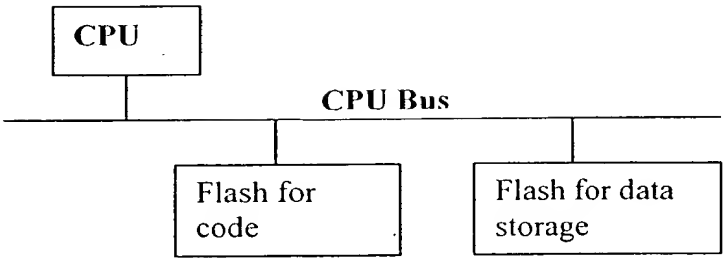


FIGURE 1 (PRIOR ART)

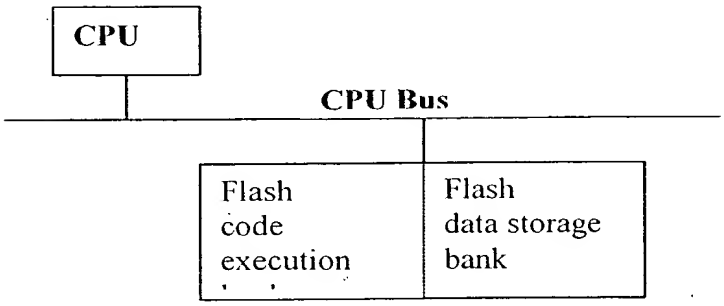


FIGURE 2 (PRIOR ART)

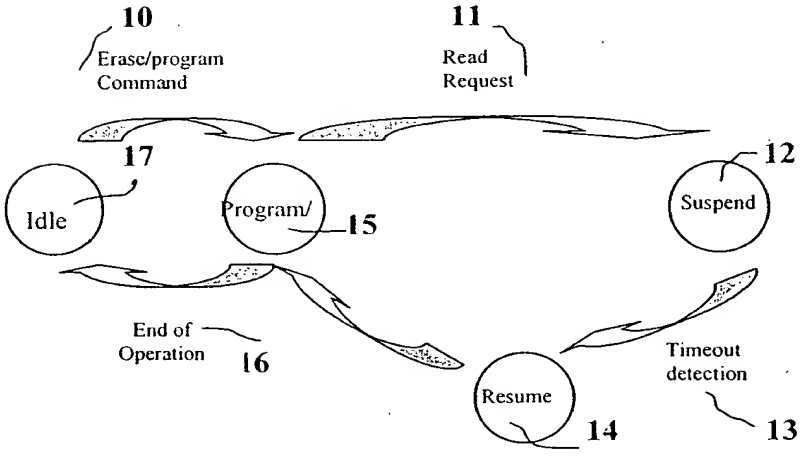


FIGURE 3

Approved MW 1/10/04 MW